

```
!pip install opencv-python scikit-image n2v csbdeep
```

```
Requirement already satisfied: opencv-python in  
/usr/local/lib/python3.11/dist-packages (4.11.0.86)
```

```
Requirement already satisfied: scikit-image in  
/usr/local/lib/python3.11/dist-packages (0.25.2)
```

```
Collecting n2v
```

```
  Downloading n2v-0.3.3-py2.py3-none-any.whl.metadata (8.6 kB)
```

```
Collecting csbdeep
```

```
  Downloading csbdeep-0.8.1-py2.py3-none-any.whl.metadata (2.4 kB)
```

```
Requirement already satisfied: numpy>=1.21.2 in  
/usr/local/lib/python3.11/dist-packages (from opencv-python) (2.0.2)
```

```
Requirement already satisfied: scipy>=1.11.4 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (1.15.2)
```

```
Requirement already satisfied: networkx>=3.0 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (3.4.2)
```

```
Requirement already satisfied: pillow>=10.1 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (11.2.1)
```

```
Requirement already satisfied: imageio!=2.35.0,>=2.33 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (2.37.0)
```

```
Requirement already satisfied: tifffile>=2022.8.12 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image)  
(2025.3.30)
```

```
Requirement already satisfied: packaging>=21 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (24.2)
```

```
Requirement already satisfied: lazy-loader>=0.4 in  
/usr/local/lib/python3.11/dist-packages (from scikit-image) (0.4)
```

```
Collecting imagecodecs>=2020.2.18 (from n2v)
```

```
  Downloading imagecodecs-2025.3.30-cp311-cp311-  
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (20 kB)
```

```
Collecting csbdeep
```

```
  Downloading csbdeep-0.7.4-py2.py3-none-any.whl.metadata (2.5 kB)
```

```
Collecting ruamel.yaml>=0.16.10 (from n2v)
```

```
  Downloading ruamel.yaml-0.18.10-py3-none-any.whl.metadata (23 kB)
```

```
Collecting bioimageio.core (from n2v)
```

```
  Downloading bioimageio_core-0.8.0-py3-none-any.whl.metadata (23 kB)
```

```
Requirement already satisfied: matplotlib in  
/usr/local/lib/python3.11/dist-packages (from csbdeep) (3.10.0)
```

```
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-  
packages (from csbdeep) (1.17.0)
```

```
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-  
packages (from csbdeep) (4.67.1)
```

```
Collecting ruamel.yaml.clib>=0.2.7 (from ruamel.yaml>=0.16.10->n2v)
```

```
  Downloading ruamel.yaml.clib-0.2.12-cp311-cp311-  
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.7 kB)
```

```
Collecting bioimageio.spec==0.5.4.1 (from bioimageio.core->n2v)
```

```
  Downloading bioimageio.spec-0.5.4.1-py3-none-any.whl.metadata (11  
kB)
```

```
Requirement already satisfied: h5py in /usr/local/lib/python3.11/dist-  
packages (from bioimageio.core->n2v) (3.13.0)
```

```

Collecting loguru (from bioimageio.core->n2v)
  Downloading loguru-0.7.3-py3-none-any.whl.metadata (22 kB)
Collecting pydantic-settings<3,>=2.5 (from bioimageio.core->n2v)
  Downloading pydantic_settings-2.9.1-py3-none-any.whl.metadata (3.8
kB)
Requirement already satisfied: pydantic<3,>=2.7.0 in
/usr/local/lib/python3.11/dist-packages (from bioimageio.core->n2v)
(2.11.3)
Requirement already satisfied: requests in
/usr/local/lib/python3.11/dist-packages (from bioimageio.core->n2v)
(2.32.3)
Collecting ruyaml (from bioimageio.core->n2v)
  Downloading ruyaml-0.91.0-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.11/dist-packages (from bioimageio.core->n2v)
(4.13.2)
Collecting xarray<2025.3.0,>=2023.01 (from bioimageio.core->n2v)
  Downloading xarray-2025.1.2-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: annotated-types<1,>=0.5.0 in
/usr/local/lib/python3.11/dist-packages (from
bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (0.7.0)
Collecting email-validator (from bioimageio.spec==0.5.4.1-
>bioimageio.core->n2v)
  Downloading email_validator-2.2.0-py3-none-any.whl.metadata (25 kB)
Requirement already satisfied: markdown in
/usr/local/lib/python3.11/dist-packages (from
bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (3.8)
Requirement already satisfied: pooch<2,>=1.5 in
/usr/local/lib/python3.11/dist-packages (from
bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (1.8.2)
Collecting pydantic<3,>=2.7.0 (from bioimageio.core->n2v)
  Downloading pydantic-2.9.2-py3-none-any.whl.metadata (149 kB)


---


149.4/149.4 kB 4.3 MB/s eta
0:00:00
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.11/dist-packages (from
bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (2.9.0.post0)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-
packages (from bioimageio.spec==0.5.4.1->bioimageio.core->n2v)
(13.9.4)
Requirement already satisfied: zipp in /usr/local/lib/python3.11/dist-
packages (from bioimageio.spec==0.5.4.1->bioimageio.core->n2v)
(3.21.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->csbdeep)
(1.3.2)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib->csbdeep)
(0.12.1)

```

Requirement already satisfied: fonttools>=4.22.0 in  
/usr/local/lib/python3.11/dist-packages (from matplotlib->csbdeep)  
(4.57.0)

Requirement already satisfied: kiwisolver>=1.3.1 in  
/usr/local/lib/python3.11/dist-packages (from matplotlib->csbdeep)  
(1.4.8)

Requirement already satisfied: pyparsing>=2.3.1 in  
/usr/local/lib/python3.11/dist-packages (from matplotlib->csbdeep)  
(3.2.3)

Collecting pydantic-core==2.23.4 (from pydantic<3,>=2.7.0-  
>bioimageio.core->n2v)

Downloading pydantic\_core-2.23.4-cp311-cp311-  
manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata (6.6 kB)

Collecting python-dotenv>=0.21.0 (from pydantic-settings<3,>=2.5-  
>bioimageio.core->n2v)

Downloading python\_dotenv-1.1.0-py3-none-any.whl.metadata (24 kB)

Requirement already satisfied: typing-inspection>=0.4.0 in  
/usr/local/lib/python3.11/dist-packages (from pydantic-  
settings<3,>=2.5->bioimageio.core->n2v) (0.4.0)

Requirement already satisfied: pandas>=2.1 in  
/usr/local/lib/python3.11/dist-packages (from  
xarray<2025.3.0,>=2023.01->bioimageio.core->n2v) (2.2.2)

Requirement already satisfied: charset-normalizer<4,>=2 in  
/usr/local/lib/python3.11/dist-packages (from requests-  
>bioimageio.core->n2v) (3.4.1)

Requirement already satisfied: idna<4,>=2.5 in  
/usr/local/lib/python3.11/dist-packages (from requests-  
>bioimageio.core->n2v) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in  
/usr/local/lib/python3.11/dist-packages (from requests-  
>bioimageio.core->n2v) (2.4.0)

Requirement already satisfied: certifi>=2017.4.17 in  
/usr/local/lib/python3.11/dist-packages (from requests-  
>bioimageio.core->n2v) (2025.4.26)

Requirement already satisfied: distro>=1.3.0 in  
/usr/local/lib/python3.11/dist-packages (from ruyaml->bioimageio.core-  
>n2v) (1.9.0)

Requirement already satisfied: setuptools>=39.0 in  
/usr/local/lib/python3.11/dist-packages (from ruyaml->bioimageio.core-  
>n2v) (75.2.0)

Requirement already satisfied: pytz>=2020.1 in  
/usr/local/lib/python3.11/dist-packages (from pandas>=2.1-  
>xarray<2025.3.0,>=2023.01->bioimageio.core->n2v) (2025.2)

Requirement already satisfied: tzdata>=2022.7 in  
/usr/local/lib/python3.11/dist-packages (from pandas>=2.1-  
>xarray<2025.3.0,>=2023.01->bioimageio.core->n2v) (2025.2)

Requirement already satisfied: platformdirs>=2.5.0 in  
/usr/local/lib/python3.11/dist-packages (from pooch<2,>=1.5-  
>bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (4.3.7)

```

Collecting dnspython>=2.0.0 (from email-validator-
>bioimageio.spec==0.5.4.1->bioimageio.core->n2v)
  Downloading dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/dist-packages (from rich-
>bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/dist-packages (from rich-
>bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-
>rich->bioimageio.spec==0.5.4.1->bioimageio.core->n2v) (0.1.2)
Downloading n2v-0.3.3-py2.py3-none-any.whl (48 kB)
_____ 48.8/48.8 kB 2.9 MB/s eta
0:00:00
_____ 69.8/69.8 kB 4.5 MB/s eta
0:00:00
agecodecs-2025.3.30-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (45.6 MB)
_____ 45.6/45.6 MB 20.0 MB/s eta
0:00:00
el.yaml-0.18.10-py3-none-any.whl (117 kB)
_____ 117.7/117.7 kB 8.5 MB/s eta
0:00:00
ageio_core-0.8.0-py3-none-any.whl (174 kB)
_____ 174.7/174.7 kB 10.7 MB/s eta
0:00:00
ageio.spec-0.5.4.1-py3-none-any.whl (210 kB)
_____ 210.3/210.3 kB 15.3 MB/s eta
0:00:00
_____ 434.9/434.9 kB 24.0 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.1 MB)
_____ 2.1/2.1 MB 61.0 MB/s eta
0:00:00
_____ 44.4/44.4 kB 2.6 MB/s eta
0:00:00
el.yaml.lib-0.2.12-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (739 kB)
_____ 739.1/739.1 kB 33.1 MB/s eta
0:00:00
_____ 1.2/1.2 MB 35.8 MB/s eta
0:00:00
_____ 61.6/61.6 kB 3.6 MB/s eta
0:00:00
l-0.91.0-py3-none-any.whl (108 kB)
_____ 108.9/108.9 kB 7.0 MB/s eta
0:00:00
ail_validator-2.2.0-py3-none-any.whl (33 kB)

```

```
Downloading dnspython-2.7.0-py3-none-any.whl (313 kB)
313.6/313.6 kB 20.2 MB/s eta
```

```
0:00:00
```

```
l, ruamel.yaml.clib, python-dotenv, pydantic-core, loguru,
imagecodecs, dnspython, ruamel.yaml, pydantic, email-validator,
xarray, pydantic-settings, csbdeep, bioimageio.spec, bioimageio.core,
n2v
```

```
Attempting uninstall: pydantic-core
```

```
Found existing installation: pydantic_core 2.33.1
```

```
Uninstalling pydantic_core-2.33.1:
```

```
Successfully uninstalled pydantic_core-2.33.1
```

```
Attempting uninstall: pydantic
```

```
Found existing installation: pydantic 2.11.3
```

```
Uninstalling pydantic-2.11.3:
```

```
Successfully uninstalled pydantic-2.11.3
```

```
Attempting uninstall: xarray
```

```
Found existing installation: xarray 2025.3.1
```

```
Uninstalling xarray-2025.3.1:
```

```
Successfully uninstalled xarray-2025.3.1
```

```
Successfully installed bioimageio.core-0.8.0 bioimageio.spec-0.5.4.1
csbdeep-0.7.4 dnspython-2.7.0 email-validator-2.2.0 imagecodecs-
2025.3.30 loguru-0.7.3 n2v-0.3.3 pydantic-2.9.2 pydantic-core-2.23.4
pydantic-settings-2.9.1 python-dotenv-1.1.0 ruamel.yaml-0.18.10
ruamel.yaml.clib-0.2.12 ruyaml-0.91.0 xarray-2025.1.2
```

```
!pip install PyWavelets # Install the missing PyWavelets package
```

```
Collecting PyWavelets
```

```
Downloading pywavelets-1.8.0-cp311-cp311-
```

```
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.0 kB)
```

```
Requirement already satisfied: numpy<3,>=1.23 in
```

```
/usr/local/lib/python3.11/dist-packages (from PyWavelets) (2.0.2)
```

```
Downloading pywavelets-1.8.0-cp311-cp311-
```

```
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.5 MB)
```

```
4.5/4.5 MB 33.4 MB/s eta
```

```
0:00:00
```

```
# Install required packages
```

```
!pip install scikit-image opencv-python-headless tensorflow matplotlib
--quiet
```

```
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_float
from skimage.restoration import denoise_wavelet
from skimage.metrics import peak_signal_noise_ratio,
structural_similarity, mean_squared_error
from tensorflow.keras.models import Model
```

```

from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
UpSampling2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing.image import img_to_array,
array_to_img

# Create folders
os.makedirs("original_images", exist_ok=True)
os.makedirs("denoised_results", exist_ok=True)

# Utility: Add synthetic noise to image
def add_noise(img, noise_type="gaussian"):
    row, col, ch = img.shape
    if noise_type == "gaussian":
        mean = 0
        sigma = 25
        gauss = np.random.normal(mean, sigma, (row, col,
ch)).reshape(row, col, ch)
        noisy = img + gauss
        return np.clip(noisy, 0, 255).astype(np.uint8)
    return img

# Utility: Denoising Autoencoder
def build_denoising_autoencoder(input_shape):
    input_img = Input(shape=input_shape)
    x = Conv2D(32, (3, 3), activation='relu', padding='same')(
input_img)
    x = MaxPooling2D((2, 2), padding='same')(x)
    x = Conv2D(16, (3, 3), activation='relu', padding='same')(x)
    encoded = MaxPooling2D((2, 2), padding='same')(x)

    x = Conv2D(16, (3, 3), activation='relu', padding='same')(encoded)
    x = UpSampling2D((2, 2))(x)
    x = Conv2D(32, (3, 3), activation='relu', padding='same')(x)
    x = UpSampling2D((2, 2))(x)
    decoded = Conv2D(3, (3, 3), activation='sigmoid', padding='same')(
x)

    autoencoder = Model(input_img, decoded)
    autoencoder.compile(optimizer=Adam(), loss='mean_squared_error')
    return autoencoder

# Main processing function
def denoise_and_compare(image_path):
    img = cv2.imread(image_path)
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img_resized = cv2.resize(img_rgb, (128, 128)) # Resize for DAE
    noisy_img = add_noise(img_resized).astype(np.uint8)

    # Convert to float

```

```

img_float = img_as_float(noisy_img)

# 1. Median Filter
median = cv2.medianBlur(noisy_img, 3)

# 2. Wavelet Denoising
wavelet = denoise_wavelet(img_float, channel_axis=-1,
rescale_sigma=True)
wavelet_uint8 = (np.clip(wavelet, 0, 1) * 255).astype(np.uint8)

# 3. Denoising Autoencoder
x_train = np.array([noisy_img]) / 255.0
y_train = np.array([img_resized]) / 255.0
dae = build_denoising_autoencoder(input_shape=(128, 128, 3))
dae.fit(x_train, y_train, epochs=100, verbose=0)
dae_output = dae.predict(x_train)[0]
dae_output_uint8 = (dae_output * 255).astype(np.uint8)

# Save outputs
cv2.imwrite("denoised_results/original.jpg",
cv2.cvtColor(img_resized, cv2.COLOR_RGB2BGR))
cv2.imwrite("denoised_results/noisy.jpg", cv2.cvtColor(noisy_img,
cv2.COLOR_RGB2BGR))
cv2.imwrite("denoised_results/median.jpg", cv2.cvtColor(median,
cv2.COLOR_RGB2BGR))
cv2.imwrite("denoised_results/wavelet.jpg",
cv2.cvtColor(wavelet_uint8, cv2.COLOR_RGB2BGR))
cv2.imwrite("denoised_results/dae.jpg",
cv2.cvtColor(dae_output_uint8, cv2.COLOR_RGB2BGR))

# Metric calculations
def metrics(original, filtered):
    return {
        "PSNR": peak_signal_noise_ratio(original, filtered),
        "SSIM": structural_similarity(original, filtered,
channel_axis=2),
        "MSE": mean_squared_error(original, filtered)
    }

print("=== Image Denoising Metrics ===")
print("Median Filter:", metrics(img_resized, median))
print("Wavelet Filter:", metrics(img_resized, wavelet_uint8))
print("Denoising Autoencoder:", metrics(img_resized,
dae_output_uint8))

# Upload and process one image
from google.colab import files
uploaded = files.upload()
import shutil
for filename in uploaded:

```

```

shutil.move(filename, f"original_images/{filename}")

# Denoise the uploaded image
img_path = os.listdir("original_images")[0]
denoise_and_compare(f"original_images/{img_path}")

<IPython.core.display.HTML object>

Saving images.jpeg to images (1).jpeg
1/1 _____ 0s 237ms/step
=== Image Denoising Metrics ===
Median Filter: {'PSNR': np.float64(22.876791317061528), 'SSIM':
np.float64(0.6427764898628349), 'MSE': np.float64(335.27504475911456)}
Wavelet Filter: {'PSNR': np.float64(24.056025946788857), 'SSIM':
np.float64(0.7196228552022315), 'MSE': np.float64(255.55110677083334)}
Denoising Autoencoder: {'PSNR': np.float64(19.566608316774555),
'SSIM': np.float64(0.577526733352884), 'MSE':
np.float64(718.4880167643229)}

# Visualization of results
def show_results():
    titles = ['Original', 'Noisy', 'Median Filter', 'Wavelet
Denoising', 'Denoising Autoencoder']
    files = ['original.jpg', 'noisy.jpg', 'median.jpg', 'wavelet.jpg',
'dae.jpg']
    plt.figure(figsize=(15, 8))

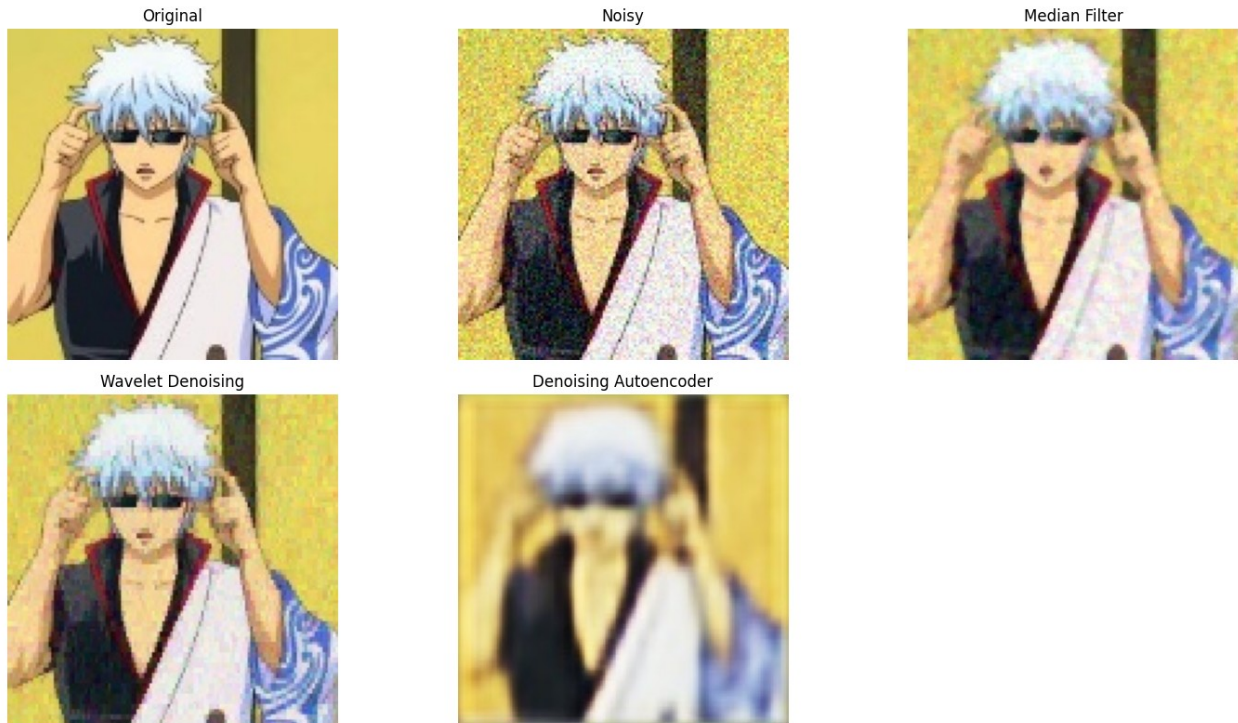
    for i, (title, file) in enumerate(zip(titles, files)):
        img = cv2.imread(f'denoised_results/{file}')
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        plt.subplot(2, 3, i+1)
        plt.imshow(img)
        plt.title(title)
        plt.axis('off')

    plt.tight_layout()
    plt.show()

# Call this function after denoising
show_results()

```





```
# Load image
import cv2
import numpy as np
import matplotlib.pyplot as plt

img_path = "/content/images.jpeg" # update with your actual filename
img = cv2.imread(img_path)
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_resized = cv2.resize(img_rgb, (128, 128)) # Resize for
autoencoder

# Add synthetic Gaussian noise
noisy_img = add_noise(img_resized).astype(np.uint8)

# Normalize and prepare input
input_img = np.expand_dims(noisy_img / 255.0, axis=0)

# Build and train autoencoder (simulating Noise2Void behavior)
autoencoder = build_denoising_autoencoder((128, 128, 3))
autoencoder.fit(input_img, input_img, epochs=100, verbose=0) # Train
on noisy image only

# Predict denoised output
denoised = autoencoder.predict(input_img)[0]
denoised_uint8 = (denoised * 255).astype(np.uint8)

# Save outputs
cv2.imwrite("denoised_results/noise2void_input.jpg",
```

```

cv2.cvtColor(noisy_img, cv2.COLOR_RGB2BGR))
cv2.imwrite("denoised_results/noise2void_output.jpg",
cv2.cvtColor(denoised_uint8, cv2.COLOR_RGB2BGR))

# Show comparison
plt.figure(figsize=(10, 4))
plt.subplot(1, 3, 1)
plt.title("Original")
plt.imshow(img_resized)
plt.axis('off')

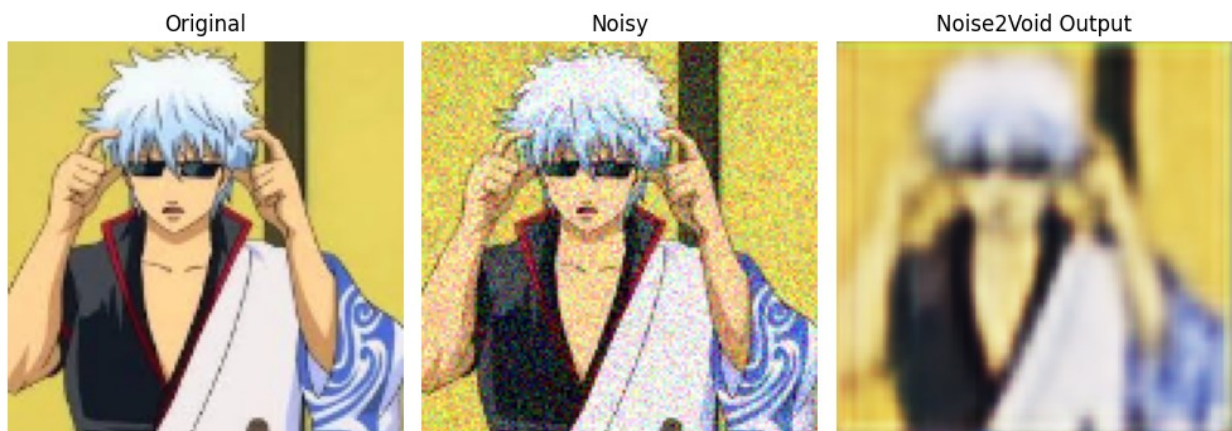
plt.subplot(1, 3, 2)
plt.title("Noisy")
plt.imshow(noisy_img)
plt.axis('off')

plt.subplot(1, 3, 3)
plt.title("Noise2Void Output")
plt.imshow(denoised_uint8)
plt.axis('off')

plt.tight_layout()
plt.show()

```

1/1 ————— 0s 204ms/step



```

# Install OpenCV if not already installed
!pip install opencv-python-headless --quiet

```

```

import cv2
import os
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files

```

```

# Step 1: Upload video

```

```

uploaded = files.upload()
video_file = list(uploaded.keys())[0]

# Create output directories
os.makedirs("frames", exist_ok=True)
os.makedirs("processed_videos", exist_ok=True)
os.makedirs("collage_frames", exist_ok=True)

# Step 2: Extract Frames from Video
cap = cv2.VideoCapture(video_file)
frame_count = 0
frames = []

while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame_path = f"frames/frame_{frame_count:04d}.jpg"
    cv2.imwrite(frame_path, frame)
    frames.append(frame)
    frame_count += 1

cap.release()
print(f"Total frames extracted: {frame_count}")

<IPython.core.display.HTML object>

Saving GxbfaBAEMAW0Xk52.mp4 to GxbfaBAEMAW0Xk52 (1).mp4
Total frames extracted: 924

# Define video writer settings
height, width, layers = frames[0].shape
fourcc = cv2.VideoWriter_fourcc(*'mp4v')

# Define output writers
out_thresh =
cv2.VideoWriter("processed_videos/adaptive_threshold.mp4", fourcc,
20.0, (width, height), False)
out_blur = cv2.VideoWriter("processed_videos/gaussian_blur.mp4",
fourcc, 20.0, (width, height), True)
out_edges = cv2.VideoWriter("processed_videos/canny_edges.mp4",
fourcc, 20.0, (width, height), False)
out_not = cv2.VideoWriter("processed_videos/bitwise_not.mp4", fourcc,
20.0, (width, height), True)

for idx, frame in enumerate(frames):
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Adaptive Thresholding
    thresh = cv2.adaptiveThreshold(gray, 255,
cv2.ADAPTIVE_THRESH_GAUSSIAN_C,

```

```

cv2.THRESH_BINARY, 11, 2)

out_thresh.write(thresh)

# Gaussian Blur
blur = cv2.GaussianBlur(frame, (15, 15), 0)
out_blur.write(blur)

# Canny Edge Detection
edges = cv2.Canny(gray, 100, 200)
out_edges.write(edges)

# Bitwise NOT
bitwise_not = cv2.bitwise_not(frame)
out_not.write(bitwise_not)

# Save some frames for collage
if idx % 20 == 0 and idx <= 100:
    cv2.imwrite(f"collage_frames/frame_{idx}.jpg", frame)

# Release video writers
out_thresh.release()
out_blur.release()
out_edges.release()
out_not.release()

print("Processed videos saved in 'processed_videos/' folder.")
Processed videos saved in 'processed_videos/' folder.

import glob

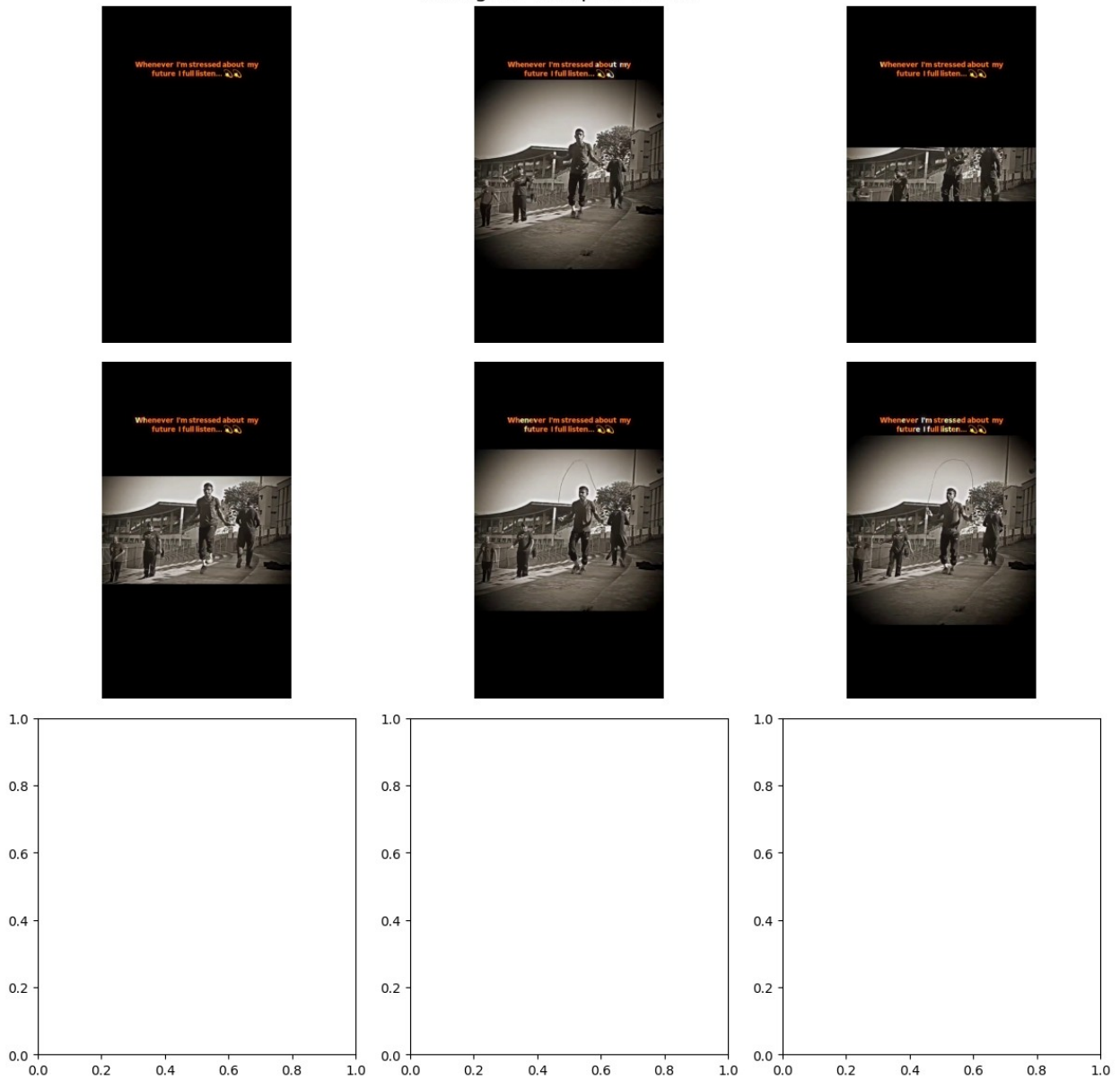
collage_images = sorted(glob.glob("collage_frames/*.jpg"))[:9]
collage_frames = [cv2.cvtColor(cv2.imread(img), cv2.COLOR_BGR2RGB) for
img in collage_images]

fig, axes = plt.subplots(3, 3, figsize=(12, 12))
for ax, img in zip(axes.flatten(), collage_frames):
    ax.imshow(img)
    ax.axis('off')

plt.suptitle("Collage of Sample Frames", fontsize=16)
plt.tight_layout()
plt.show()

```

Collage of Sample Frames



```
import kagglehub
import os

# Step 1: Download the dataset
path = kagglehub.dataset_download("pevogam/ucf101")

# Step 2: Explore the downloaded files and directories
print("Base download path:", path)
print("\nContents of the downloaded path:")
print(os.listdir(path))

# Step 3: Search for the UCF101 video classes
for root, dirs, files in os.walk(path):
```

```
print(f"\nFound directory: {root}")
for d in dirs:
    print("    └─", d)
break # only print the first level
```

Base download path: /kaggle/input/ucf101

Contents of the downloaded path:

```
['UCF101', 'UCF101TrainTestSplits-RecognitionTask']
```

Found directory: /kaggle/input/ucf101

└─ UCF101

└─ UCF101TrainTestSplits-RecognitionTask

```
import os
```

```
root_dir = "/kaggle/input/ucf101/UCF101"
```

```
for root, dirs, files in os.walk(root_dir):
    print("Directory:", root)
    for d in dirs:
        print("    Subfolder:", d)
    for f in files[:3]: # print only a few files for brevity
        print("        File:", f)
    print("-" * 40)
```

Directory: /kaggle/input/ucf101/UCF101

Subfolder: UCF-101

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101

Subfolder: HorseRace

Subfolder: StillRings

Subfolder: ApplyLipstick

Subfolder: HammerThrow

Subfolder: VolleyballSpiking

Subfolder: Biking

Subfolder: PlayingCello

Subfolder: BodyWeightSquats

Subfolder: TaiChi

Subfolder: Punch

Subfolder: BreastStroke

Subfolder: Billiards

Subfolder: BoxingPunchingBag

Subfolder: BasketballDunk

Subfolder: PoleVault

Subfolder: ThrowDiscus

Subfolder: BaseballPitch

Subfolder: Knitting

Subfolder: SumoWrestling

Subfolder: HorseRiding

Subfolder: Mixing

Subfolder: BrushingTeeth  
Subfolder: HighJump  
Subfolder: Skijet  
Subfolder: SkateBoarding  
Subfolder: MilitaryParade  
Subfolder: IceDancing  
Subfolder: CricketShot  
Subfolder: Fencing  
Subfolder: JugglingBalls  
Subfolder: Swing  
Subfolder: RockClimbingIndoor  
Subfolder: PlayingFlute  
Subfolder: SalsaSpin  
Subfolder: CricketBowling  
Subfolder: Typing  
Subfolder: ApplyEyeMakeup  
Subfolder: PlayingTabla  
Subfolder: BalanceBeam  
Subfolder: FloorGymnastics  
Subfolder: HeadMassage  
Subfolder: FrisbeeCatch  
Subfolder: Rowing  
Subfolder: Hammering  
Subfolder: CuttingInKitchen  
Subfolder: BenchPress  
Subfolder: PushUps  
Subfolder: Nunchucks  
Subfolder: Archery  
Subfolder: LongJump  
Subfolder: BlowingCandles  
Subfolder: WallPushups  
Subfolder: PlayingViolin  
Subfolder: PullUps  
Subfolder: PlayingPiano  
Subfolder: PlayingDhol  
Subfolder: FrontCrawl  
Subfolder: HulaHoop  
Subfolder: CliffDiving  
Subfolder: BandMarching  
Subfolder: ParallelBars  
Subfolder: RopeClimbing  
Subfolder: YoYo  
Subfolder: TrampolineJumping  
Subfolder: JumpingJack  
Subfolder: Drumming  
Subfolder: TableTennisShot  
Subfolder: BabyCrawling  
Subfolder: GolfSwing  
Subfolder: PlayingGuitar

Subfolder: BlowDryHair  
Subfolder: PizzaTossing  
Subfolder: SoccerPenalty  
Subfolder: Lunges  
Subfolder: TennisSwing  
Subfolder: WalkingWithDog  
Subfolder: ShavingBeard  
Subfolder: PlayingDaf  
Subfolder: Bowling  
Subfolder: Surfing  
Subfolder: Kayaking  
Subfolder: PommelHorse  
Subfolder: Basketball  
Subfolder: SkyDiving  
Subfolder: Rafting  
Subfolder: JumpRope  
Subfolder: HandstandWalking  
Subfolder: FieldHockeyPenalty  
Subfolder: CleanAndJerk  
Subfolder: UnevenBars  
Subfolder: Skiing  
Subfolder: Shotput  
Subfolder: Diving  
Subfolder: HandstandPushups  
Subfolder: Haircut  
Subfolder: PlayingSitar  
Subfolder: BoxingSpeedBag  
Subfolder: SoccerJuggling  
Subfolder: MoppingFloor  
Subfolder: JavelinThrow  
Subfolder: WritingOnBoard

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HorseRace  
File: v\_HorseRace\_g23\_c05.avi  
File: v\_HorseRace\_g18\_c05.avi  
File: v\_HorseRace\_g10\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/StillRings  
File: v\_StillRings\_g21\_c05.avi  
File: v\_StillRings\_g23\_c02.avi  
File: v\_StillRings\_g08\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/ApplyLipstick  
File: v\_ApplyLipstick\_g06\_c05.avi  
File: v\_ApplyLipstick\_g04\_c03.avi  
File: v\_ApplyLipstick\_g03\_c04.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HammerThrow  
File: v\_HammerThrow\_g17\_c04.avi



File: v\_HammerThrow\_g11\_c01.avi

File: v\_HammerThrow\_g01\_c06.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/VolleyballSpiking

File: v\_VolleyballSpiking\_g05\_c02.avi

File: v\_VolleyballSpiking\_g05\_c04.avi

File: v\_VolleyballSpiking\_g02\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Biking

File: v\_Biking\_g03\_c03.avi

File: v\_Biking\_g07\_c05.avi

File: v\_Biking\_g17\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingCello

File: v\_PlayingCello\_g08\_c04.avi

File: v\_PlayingCello\_g11\_c01.avi

File: v\_PlayingCello\_g22\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BodyWeightSquats

File: v\_BodyWeightSquats\_g12\_c01.avi

File: v\_BodyWeightSquats\_g25\_c05.avi

File: v\_BodyWeightSquats\_g11\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/TaiChi

File: v\_TaiChi\_g09\_c02.avi

File: v\_TaiChi\_g13\_c02.avi

File: v\_TaiChi\_g10\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Punch

File: v\_Punch\_g02\_c01.avi

File: v\_Punch\_g16\_c02.avi

File: v\_Punch\_g04\_c04.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BreastStroke

File: v\_BreastStroke\_g05\_c02.avi

File: v\_BreastStroke\_g04\_c04.avi

File: v\_BreastStroke\_g11\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Billiards

File: v\_Billiards\_g08\_c01.avi

File: v\_Billiards\_g01\_c06.avi

File: v\_Billiards\_g05\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BoxingPunchingBag

File: v\_BoxingPunchingBag\_g01\_c05.avi

File: v\_BoxingPunchingBag\_g22\_c07.avi

File: v\_BoxingPunchingBag\_g25\_c04.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BasketballDunk

File: v\_BasketballDunk\_g18\_c03.avi  
File: v\_BasketballDunk\_g15\_c03.avi  
File: v\_BasketballDunk\_g01\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PoleVault

File: v\_PoleVault\_g12\_c01.avi  
File: v\_PoleVault\_g10\_c07.avi  
File: v\_PoleVault\_g13\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/ThrowDiscus

File: v\_ThrowDiscus\_g22\_c01.avi  
File: v\_ThrowDiscus\_g07\_c04.avi  
File: v\_ThrowDiscus\_g25\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BaseballPitch

File: v\_BaseballPitch\_g10\_c05.avi  
File: v\_BaseballPitch\_g22\_c03.avi  
File: v\_BaseballPitch\_g08\_c06.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Knitting

File: v\_Knitting\_g23\_c01.avi  
File: v\_Knitting\_g07\_c03.avi  
File: v\_Knitting\_g08\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/SumoWrestling

File: v\_SumoWrestling\_g15\_c03.avi  
File: v\_SumoWrestling\_g04\_c01.avi  
File: v\_SumoWrestling\_g24\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HorseRiding

File: v\_HorseRiding\_g11\_c01.avi  
File: v\_HorseRiding\_g15\_c02.avi  
File: v\_HorseRiding\_g06\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Mixing

File: v\_Mixing\_g11\_c02.avi  
File: v\_Mixing\_g08\_c04.avi  
File: v\_Mixing\_g11\_c06.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BrushingTeeth

File: v\_BrushingTeeth\_g13\_c01.avi  
File: v\_BrushingTeeth\_g24\_c03.avi  
File: v\_BrushingTeeth\_g07\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HighJump

File: v\_HighJump\_g19\_c05.avi  
File: v\_HighJump\_g11\_c01.avi  
File: v\_HighJump\_g05\_c03.avi

```
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Skijet
File: v_Skijet_g17_c02.avi
File: v_Skijet_g23_c01.avi
File: v_Skijet_g16_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/SkateBoarding
File: v_SkateBoarding_g16_c02.avi
File: v_SkateBoarding_g07_c01.avi
File: v_SkateBoarding_g12_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/MilitaryParade
File: v_MilitaryParade_g20_c04.avi
File: v_MilitaryParade_g16_c03.avi
File: v_MilitaryParade_g19_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/IceDancing
File: v_IceDancing_g04_c07.avi
File: v_IceDancing_g05_c03.avi
File: v_IceDancing_g16_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/CricketShot
File: v_CricketShot_g21_c06.avi
File: v_CricketShot_g19_c02.avi
File: v_CricketShot_g18_c05.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Fencing
File: v_Fencing_g20_c02.avi
File: v_Fencing_g21_c01.avi
File: v_Fencing_g10_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/JugglingBalls
File: v_JugglingBalls_g18_c04.avi
File: v_JugglingBalls_g17_c04.avi
File: v_JugglingBalls_g09_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Swing
File: v_Swing_g04_c04.avi
File: v_Swing_g04_c03.avi
File: v_Swing_g17_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/RockClimbingIndoor
File: v_RockClimbingIndoor_g03_c05.avi
File: v_RockClimbingIndoor_g15_c03.avi
File: v_RockClimbingIndoor_g16_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingFlute
File: v_PlayingFlute_g21_c04.avi
File: v_PlayingFlute_g03_c04.avi
File: v_PlayingFlute_g23_c03.avi
```

```
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/SalsaSpin  
File: v_SalsaSpin_g12_c02.avi  
File: v_SalsaSpin_g18_c04.avi  
File: v_SalsaSpin_g09_c01.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/CricketBowling  
File: v_CricketBowling_g08_c02.avi  
File: v_CricketBowling_g24_c03.avi  
File: v_CricketBowling_g24_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Typing  
File: v_Typing_g01_c06.avi  
File: v_Typing_g23_c02.avi  
File: v_Typing_g20_c05.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/ApplyEyeMakeup  
File: v_ApplyEyeMakeup_g19_c01.avi  
File: v_ApplyEyeMakeup_g25_c05.avi  
File: v_ApplyEyeMakeup_g13_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingTabla  
File: v_PlayingTabla_g18_c05.avi  
File: v_PlayingTabla_g17_c03.avi  
File: v_PlayingTabla_g01_c01.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BalanceBeam  
File: v_BalanceBeam_g21_c02.avi  
File: v_BalanceBeam_g10_c04.avi  
File: v_BalanceBeam_g18_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/FloorGymnastics  
File: v_FloorGymnastics_g12_c04.avi  
File: v_FloorGymnastics_g18_c03.avi  
File: v_FloorGymnastics_g08_c03.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HeadMassage  
File: v_HeadMassage_g06_c04.avi  
File: v_HeadMassage_g19_c02.avi  
File: v_HeadMassage_g04_c03.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/FrisbeeCatch  
File: v_FrisbeeCatch_g24_c02.avi  
File: v_FrisbeeCatch_g16_c04.avi  
File: v_FrisbeeCatch_g01_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Rowing  
File: v_Rowing_g08_c04.avi  
File: v_Rowing_g19_c01.avi
```

File: v\_Rowing\_g18\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Hammering

File: v\_Hammering\_g17\_c02.avi

File: v\_Hammering\_g17\_c06.avi

File: v\_Hammering\_g09\_c06.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/CuttingInKitchen

File: v\_CuttingInKitchen\_g13\_c02.avi

File: v\_CuttingInKitchen\_g16\_c02.avi

File: v\_CuttingInKitchen\_g20\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BenchPress

File: v\_BenchPress\_g19\_c02.avi

File: v\_BenchPress\_g12\_c04.avi

File: v\_BenchPress\_g07\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PushUps

File: v\_PushUps\_g18\_c02.avi

File: v\_PushUps\_g22\_c04.avi

File: v\_PushUps\_g03\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Nunchucks

File: v\_Nunchucks\_g03\_c05.avi

File: v\_Nunchucks\_g17\_c06.avi

File: v\_Nunchucks\_g18\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Archery

File: v\_Archery\_g25\_c05.avi

File: v\_Archery\_g16\_c02.avi

File: v\_Archery\_g20\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/LongJump

File: v\_LongJump\_g07\_c01.avi

File: v\_LongJump\_g03\_c06.avi

File: v\_LongJump\_g10\_c06.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BlowingCandles

File: v\_BlowingCandles\_g06\_c05.avi

File: v\_BlowingCandles\_g12\_c02.avi

File: v\_BlowingCandles\_g11\_c04.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/WallPushups

File: v\_WallPushups\_g10\_c04.avi

File: v\_WallPushups\_g16\_c07.avi

File: v\_WallPushups\_g08\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingViolin

File: v\_PlayingViolin\_g18\_c01.avi

File: v\_PlayingViolin\_g12\_c03.avi

File: v\_PlayingViolin\_g16\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PullUps

File: v\_PullUps\_g15\_c03.avi

File: v\_PullUps\_g20\_c04.avi

File: v\_PullUps\_g13\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingPiano

File: v\_PlayingPiano\_g22\_c02.avi

File: v\_PlayingPiano\_g16\_c06.avi

File: v\_PlayingPiano\_g22\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingDhol

File: v\_PlayingDhol\_g06\_c05.avi

File: v\_PlayingDhol\_g04\_c04.avi

File: v\_PlayingDhol\_g18\_c02.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/FrontCrawl

File: v\_FrontCrawl\_g14\_c01.avi

File: v\_FrontCrawl\_g04\_c02.avi

File: v\_FrontCrawl\_g17\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HulaHoop

File: v\_HulaHoop\_g12\_c06.avi

File: v\_HulaHoop\_g12\_c03.avi

File: v\_HulaHoop\_g05\_c04.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/CliffDiving

File: v\_CliffDiving\_g12\_c03.avi

File: v\_CliffDiving\_g19\_c06.avi

File: v\_CliffDiving\_g19\_c03.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BandMarching

File: v\_BandMarching\_g01\_c02.avi

File: v\_BandMarching\_g02\_c03.avi

File: v\_BandMarching\_g08\_c07.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/ParallelBars

File: v\_ParallelBars\_g10\_c02.avi

File: v\_ParallelBars\_g11\_c01.avi

File: v\_ParallelBars\_g01\_c01.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/RopeClimbing

File: v\_RopeClimbing\_g22\_c03.avi

File: v\_RopeClimbing\_g06\_c01.avi

File: v\_RopeClimbing\_g13\_c07.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/YoYo

File: v\_YoYo\_g01\_c01.avi  
File: v\_YoYo\_g19\_c04.avi  
File: v\_YoYo\_g13\_c05.avi

-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/TrampolineJumping  
File: v\_TrampolineJumping\_g02\_c05.avi  
File: v\_TrampolineJumping\_g14\_c04.avi  
File: v\_TrampolineJumping\_g10\_c06.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/JumpingJack  
File: v\_JumpingJack\_g08\_c03.avi  
File: v\_JumpingJack\_g19\_c06.avi  
File: v\_JumpingJack\_g11\_c02.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/Drumming  
File: v\_Drumming\_g10\_c02.avi  
File: v\_Drumming\_g15\_c03.avi  
File: v\_Drumming\_g17\_c01.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/TableTennisShot  
File: v\_TableTennisShot\_g03\_c05.avi  
File: v\_TableTennisShot\_g20\_c03.avi  
File: v\_TableTennisShot\_g05\_c06.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/BabyCrawling  
File: v\_BabyCrawling\_g02\_c06.avi  
File: v\_BabyCrawling\_g04\_c01.avi  
File: v\_BabyCrawling\_g17\_c02.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/GolfSwing  
File: v\_GolfSwing\_g17\_c06.avi  
File: v\_GolfSwing\_g02\_c04.avi  
File: v\_GolfSwing\_g04\_c03.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingGuitar  
File: v\_PlayingGuitar\_g09\_c02.avi  
File: v\_PlayingGuitar\_g11\_c07.avi  
File: v\_PlayingGuitar\_g08\_c01.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/BlowDryHair  
File: v\_BlowDryHair\_g16\_c01.avi  
File: v\_BlowDryHair\_g19\_c04.avi  
File: v\_BlowDryHair\_g03\_c03.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/PizzaTossing  
File: v\_PizzaTossing\_g12\_c02.avi  
File: v\_PizzaTossing\_g09\_c02.avi  
File: v\_PizzaTossing\_g11\_c03.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/SoccerPenalty  
File: v\_SoccerPenalty\_g06\_c05.avi  
File: v\_SoccerPenalty\_g21\_c05.avi  
File: v\_SoccerPenalty\_g10\_c03.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/Lunges  
File: v\_Lunges\_g01\_c07.avi  
File: v\_Lunges\_g17\_c04.avi  
File: v\_Lunges\_g12\_c04.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/TennisSwing  
File: v\_TennisSwing\_g08\_c06.avi  
File: v\_TennisSwing\_g10\_c02.avi  
File: v\_TennisSwing\_g04\_c06.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/WalkingWithDog  
File: v\_WalkingWithDog\_g13\_c02.avi  
File: v\_WalkingWithDog\_g09\_c02.avi  
File: v\_WalkingWithDog\_g09\_c03.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/ShavingBeard  
File: v\_ShavingBeard\_g07\_c02.avi  
File: v\_ShavingBeard\_g01\_c03.avi  
File: v\_ShavingBeard\_g14\_c01.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingDaf  
File: v\_PlayingDaf\_g03\_c04.avi  
File: v\_PlayingDaf\_g15\_c05.avi  
File: v\_PlayingDaf\_g16\_c05.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/Bowling  
File: v\_Bowling\_g11\_c02.avi  
File: v\_Bowling\_g17\_c02.avi  
File: v\_Bowling\_g08\_c02.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/Surfing  
File: v\_Surfing\_g21\_c02.avi  
File: v\_Surfing\_g01\_c06.avi  
File: v\_Surfing\_g17\_c06.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/Kayaking  
File: v\_Kayaking\_g08\_c06.avi  
File: v\_Kayaking\_g11\_c07.avi  
File: v\_Kayaking\_g22\_c02.avi  
-----

Directory: /kaggle/input/ucf101/UCF101/UCF-101/PommelHorse  
File: v\_PommelHorse\_g08\_c04.avi  
File: v\_PommelHorse\_g07\_c05.avi  
File: v\_PommelHorse\_g08\_c03.avi  
-----



```
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Basketball
File: v_Basketball_g19_c04.avi
File: v_Basketball_g20_c01.avi
File: v_Basketball_g15_c03.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/SkyDiving
File: v_SkyDiving_g08_c04.avi
File: v_SkyDiving_g19_c03.avi
File: v_SkyDiving_g17_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Rafting
File: v_Rafting_g18_c04.avi
File: v_Rafting_g17_c04.avi
File: v_Rafting_g14_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/JumpRope
File: v_JumpRope_g04_c06.avi
File: v_JumpRope_g06_c04.avi
File: v_JumpRope_g23_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HandstandWalking
File: v_HandstandWalking_g02_c01.avi
File: v_HandstandWalking_g02_c04.avi
File: v_HandstandWalking_g07_c02.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/FieldHockeyPenalty
File: v_FieldHockeyPenalty_g13_c03.avi
File: v_FieldHockeyPenalty_g22_c02.avi
File: v_FieldHockeyPenalty_g08_c02.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/CleanAndJerk
File: v_CleanAndJerk_g04_c02.avi
File: v_CleanAndJerk_g04_c04.avi
File: v_CleanAndJerk_g12_c02.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/UnevenBars
File: v_UnevenBars_g02_c03.avi
File: v_UnevenBars_g24_c01.avi
File: v_UnevenBars_g11_c01.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Skiing
File: v_Skiing_g16_c04.avi
File: v_Skiing_g15_c01.avi
File: v_Skiing_g07_c04.avi
-----
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Shotput
File: v_Shotput_g19_c01.avi
File: v_Shotput_g17_c01.avi
File: v_Shotput_g10_c06.avi
```

```
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Diving  
File: v_Diving_g13_c04.avi  
File: v_Diving_g04_c07.avi  
File: v_Diving_g21_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/HandstandPushups  
File: v_HandStandPushups_g11_c03.avi  
File: v_HandStandPushups_g01_c04.avi  
File: v_HandStandPushups_g21_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/Haircut  
File: v_Haircut_g07_c03.avi  
File: v_Haircut_g20_c02.avi  
File: v_Haircut_g03_c06.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/PlayingSitar  
File: v_PlayingSitar_g10_c01.avi  
File: v_PlayingSitar_g08_c03.avi  
File: v_PlayingSitar_g01_c04.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/BoxingSpeedBag  
File: v_BoxingSpeedBag_g03_c05.avi  
File: v_BoxingSpeedBag_g23_c03.avi  
File: v_BoxingSpeedBag_g19_c03.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/SoccerJuggling  
File: v_SoccerJuggling_g24_c03.avi  
File: v_SoccerJuggling_g04_c01.avi  
File: v_SoccerJuggling_g12_c02.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/MoppingFloor  
File: v_MoppingFloor_g25_c04.avi  
File: v_MoppingFloor_g20_c03.avi  
File: v_MoppingFloor_g11_c04.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/JavelinThrow  
File: v_JavelinThrow_g06_c04.avi  
File: v_JavelinThrow_g21_c04.avi  
File: v_JavelinThrow_g07_c01.avi  
-----  
Directory: /kaggle/input/ucf101/UCF101/UCF-101/WritingOnBoard  
File: v_WritingOnBoard_g16_c04.avi  
File: v_WritingOnBoard_g19_c04.avi  
File: v_WritingOnBoard_g12_c02.avi  
-----
```

```
import os  
import shutil  
import random
```

```

# Define source and destination directories
SOURCE_DIR = '/kaggle/input/ucf101/UCF101/UCF-101'
DEST_DIR = '/kaggle/working/UCF101_subset'

# List of selected classes (can be updated as needed)
SELECTED_CLASSES = ['Basketball', 'Biking', 'PlayingGuitar', 'Typing',
'JumpRope']
VIDEOS_PER_CLASS = 10

# Create the destination directory if it doesn't exist
os.makedirs(DEST_DIR, exist_ok=True)

# Iterate over the selected classes and copy videos
for cls in SELECTED_CLASSES:
    class_path = os.path.join(SOURCE_DIR, cls)
    dest_class_path = os.path.join(DEST_DIR, cls)

    # Create class folder in destination
    os.makedirs(dest_class_path, exist_ok=True)

    # Select random 10 videos from the class
    selected = random.sample(os.listdir(class_path), VIDEOS_PER_CLASS)

    # Copy selected videos to the destination
    for video in selected:
        shutil.copy(os.path.join(class_path, video), dest_class_path)

print(f"Subset created at: {DEST_DIR}")

Subset created at: /kaggle/working/UCF101_subset

import cv2
import os
import numpy as np

# Define parameters
FRAME_RATE = 5 # Extract every 5th frame
RESIZE_DIM = (112, 112) # Resize frames to 112x112
MAX_FRAMES = 16 # Number of frames per video

# Function to extract frames from video
def extract_frames(video_path, max_frames=MAX_FRAMES,
frame_rate=FRAME_RATE, resize_dim=RESIZE_DIM):
    # Read the video
    cap = cv2.VideoCapture(video_path)

    frames = []
    frame_count = 0
    while True:
        ret, frame = cap.read()

```

```

        if not ret:
            break

        # Extract every 'frame_rate'-th frame
        if frame_count % frame_rate == 0:
            frame_resized = cv2.resize(frame, resize_dim)
            frames.append(frame_resized)

        frame_count += 1

        # Stop once we have extracted enough frames
        if len(frames) == max_frames:
            break

    cap.release()

    # If fewer than MAX_FRAMES are extracted, pad the sequence with the last frame
    while len(frames) < max_frames:
        frames.append(frames[-1])

    return np.array(frames)

# Process each class and video
video_frames = {}
for cls in SELECTED_CLASSES:
    class_path = os.path.join(DEST_DIR, cls)
    video_frames[cls] = []

    for video in os.listdir(class_path):
        video_path = os.path.join(class_path, video)
        frames = extract_frames(video_path)
        video_frames[cls].append(frames)

print("Frame extraction completed.")
Frame extraction completed.

from sklearn.preprocessing import LabelEncoder

# Initialize label encoder
label_encoder = LabelEncoder()

# Fit the encoder on the selected classes
labels = label_encoder.fit_transform(SELECTED_CLASSES)

# Create a dictionary to map class names to labels
class_labels = dict(zip(SELECTED_CLASSES, labels))
print("Label encoding completed.")
Label encoding completed.

```

```

from sklearn.model_selection import train_test_split

# Prepare data and labels
data = []
labels = []

# Add frames and their corresponding labels
for cls in SELECTED_CLASSES:
    for frames in video_frames[cls]:
        data.append(frames)
        labels.append(class_labels[cls])

# Convert to numpy arrays
data = np.array(data)
labels = np.array(labels)

# Split data into training and testing sets (80/20 split)
X_train, X_test, y_train, y_test = train_test_split(data, labels,
test_size=0.2, random_state=42)

print(f"Training data shape: {X_train.shape}")
print(f"Testing data shape: {X_test.shape}")

Training data shape: (40, 16, 112, 112, 3)
Testing data shape: (10, 16, 112, 112, 3)

import tensorflow as tf
from tensorflow.keras import layers, models

# Define 3D CNN Model
def create_3d_cnn_model(input_shape, num_classes):
    model = models.Sequential()

    # 3D convolution layers with padding='same'
    model.add(layers.Conv3D(32, kernel_size=(3, 3, 3),
activation='relu', input_shape=input_shape, padding='same'))
    model.add(layers.MaxPooling3D(pool_size=(2, 2, 2)))
    model.add(layers.Conv3D(64, kernel_size=(3, 3, 3),
activation='relu', padding='same'))
    model.add(layers.MaxPooling3D(pool_size=(2, 2, 2)))
    model.add(layers.Conv3D(128, kernel_size=(3, 3, 3),
activation='relu', padding='same'))
    model.add(layers.MaxPooling3D(pool_size=(2, 2, 2)))

    # Flatten and fully connected layers
    model.add(layers.Flatten())
    model.add(layers.Dense(128, activation='relu'))
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model

```

```

# Define input shape based on frame size and sequence length
input_shape = (MAX_FRAMES, RESIZE_DIM[0], RESIZE_DIM[1], 3) # (16,
112, 112, 3) for 16 frames of 112x112 images with 3 color channels
num_classes = len(SELECTED_CLASSES)

# Create the model
model = create_3d_cnn_model(input_shape, num_classes)

# Compile the model
model.compile(optimizer='adam',
loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Model summary
model.summary()

/usr/local/lib/python3.11/dist-packages/keras/src/layers/
convolutional/base_conv.py:107: UserWarning: Do not pass an
`input_shape`/`input_dim` argument to a layer. When using Sequential
models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
Model: "sequential_1"

```

Layer (type) Param #	Output Shape	
conv3d_3 (Conv3D) 2,624	(None, 16, 112, 112, 32)	
max_pooling3d_3 (MaxPooling3D) 0	(None, 8, 56, 56, 32)	
conv3d_4 (Conv3D) 55,360	(None, 8, 56, 56, 64)	
max_pooling3d_4 (MaxPooling3D) 0	(None, 4, 28, 28, 64)	
conv3d_5 (Conv3D)	(None, 4, 28, 28, 128)	

221,312				
		max_pooling3d_5 (MaxPooling3D)	(None, 2, 14, 14, 128)	
0				
		flatten_1 (Flatten)	(None, 50176)	
0				
		dense_2 (Dense)	(None, 128)	
6,422,656				
		dense_3 (Dense)	(None, 5)	
645				

Total params: 6,702,597 (25.57 MB)

Trainable params: 6,702,597 (25.57 MB)

Non-trainable params: 0 (0.00 B)

*# Train the model*

```
history = model.fit(X_train, y_train, epochs=10, batch_size=4,
validation_data=(X_test, y_test))
```

*# Save the model*

```
model.save("/kaggle/working/ucf101_3dcnn_model.h5")
```

```
print("Training completed and model saved.")
```

Epoch 1/10

```
10/10 _____ 53s 5s/step - accuracy: 0.1854 - loss:
398.4858 - val_accuracy: 0.4000 - val_loss: 1.5153
```

Epoch 2/10

```
10/10 _____ 81s 5s/step - accuracy: 0.2454 - loss:
1.5007 - val_accuracy: 0.2000 - val_loss: 2.3985
```

Epoch 3/10

```
10/10 _____ 52s 5s/step - accuracy: 0.5691 - loss:
1.3393 - val_accuracy: 0.1000 - val_loss: 1.5773
```

Epoch 4/10

```
10/10 _____ 52s 5s/step - accuracy: 0.6785 - loss:
1.1525 - val_accuracy: 0.3000 - val_loss: 2.2342
```

Epoch 5/10

```
10/10 _____ 80s 5s/step - accuracy: 0.6778 - loss:
1.4211 - val_accuracy: 0.3000 - val_loss: 2.1582
```

Epoch 6/10

```

10/10 ————— 83s 5s/step - accuracy: 0.5804 - loss:
0.9205 - val_accuracy: 0.1000 - val_loss: 1.7314
Epoch 7/10
10/10 ————— 83s 5s/step - accuracy: 0.8245 - loss:
0.6209 - val_accuracy: 0.3000 - val_loss: 1.7927
Epoch 8/10
10/10 ————— 82s 5s/step - accuracy: 0.7939 - loss:
0.5056 - val_accuracy: 0.6000 - val_loss: 2.0012
Epoch 9/10
10/10 ————— 79s 5s/step - accuracy: 0.8435 - loss:
0.4716 - val_accuracy: 0.5000 - val_loss: 1.4895
Epoch 10/10
10/10 ————— 82s 5s/step - accuracy: 0.7711 - loss:
0.5867 - val_accuracy: 0.5000 - val_loss: 1.6238

```

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save\_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my\_model.keras')` or `keras.saving.save\_model(model, 'my\_model.keras')`.

Training completed and model saved.

```

from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
import matplotlib.pyplot as plt

```

```

# Evaluate the model on the test set
y_pred = model.predict(X_test)
y_pred_classes = np.argmax(y_pred, axis=1)

```

```

# Print classification report
print("Classification Report:")
print(classification_report(y_test, y_pred_classes,
target_names=SELECTED_CLASSES))

```

```

# Compute confusion matrix
cm = confusion_matrix(y_test, y_pred_classes)

```

```

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=SELECTED_CLASSES, yticklabels=SELECTED_CLASSES)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

```

```

1/1 ————— 3s 3s/step
Classification Report:
          precision    recall  f1-score   support

```



Basketball	0.00	0.00	0.00	0
Biking	0.67	0.67	0.67	3
PlayingGuitar	0.50	0.50	0.50	2
Typing	0.50	0.50	0.50	2
JumpRope	1.00	0.33	0.50	3

accuracy			0.50	10
macro avg	0.53	0.40	0.43	10
weighted avg	0.70	0.50	0.55	10

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
```

```
_warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

